



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A
BIOMECHANIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND
BIOMECHANICS

DEFORMAČNÍ MECHANISMY V KRYSTALECH POMOCÍ MOLEKULÁRNÍ DYNAMIKY

DEFORMATION MECHANISMS IN CRYSTALS BY MEANS OF MOLECULAR DYNAMICS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VOJTĚCH LAMBERSKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Mgr. MIROSLAV ČERNÝ, Ph.D.

BRNO 2008

Zadání

Abstrakt

Tato práce se zabývá problematikou molekulární dynamiky, chováním pevných látek na atomární úrovni. V rešeršní studii je nastíněn princip výpočtů, které se používají při určování chování skupin atomů nebo molekul. Dále je popsán způsob, jakým počítá metoda EAM, jak se provede paralelizace pro více procesorů a zefektivnění výpočtu. Nakonec je proveden výpočet teoretické pevnosti krystalu v tahu pomocí programu Lammmps.

Abstract

This work deals with molecular dynamics modeling of processes in condensed matter on atomic level. The physical principles used to predict motion of atom or molecule groups are described in the retrieval part. Then follows a description of the EAM method, ways how to parallelize computing on many processors and how perform calculation optimizing. Finally, we perform a theoretical tensile strength computation using Lammmps program.

Klíčová slova

systém Lammmps, molekulární dynamika, teoretická pevnost v tahu, zobrazení atomů v prostoru, metoda vnořeného atomu, paralelní algoritmy

Key words

Lammmps software, molecular dynamics, theoretical tensile strength, atomistic configuration viewing, embeded atom method, parallel computing

Bibliografická citace

LAMBERSKÝ, V. *Deformační mechanismy v krystalech pomocí molekulární dynamiky*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2008. 48 s. Vedoucí bakalářské práce doc. Mgr. Miroslav Černý, Ph.D.

Čestné prohlášení

Čestně prohlašuji, že bakalářskou práci na téma Deformační mechanizmy v krystalech pomocí molekulární dynamiky jsem vypracoval samostatně pod vedením svého vedoucího bakalářské práce s použitím odborné literatury, kterou jsem všemnu citoval v seznamu literatury.

V Brně dne 23. 5. 2008

.....
Vojtěch Lamberský

Poděkování

Na tomto místě bych rád poděkoval svému školiteli doc. Mgr. Miroslavu Černému, Ph.D. za jeho vedení, cenné rady a připomínky během zpracování bakalářské práce a Ing. J. Reslerovi za pomoc s překladem programu.

Obsah

1	Úvod	9
2	Formulace problému a stanovení cílů	11
3	Fyzikální principy	12
3.1	Teorie vazeb	12
3.2	Elektrické pole	13
3.3	Práce	13
3.4	Potenciální energie	14
3.5	Vazby v pevných látkách	15
3.6	Kovová vazba	16
3.7	Vliv teploty	17
3.8	Odpudivé síly	18
4	Metoda vnořného atomu a postup výpočtu	19
4.1	Hlavní problémy	19
4.2	Použití	20
4.3	Princip výpočtu	20
4.4	Paralelní algoritmy	21
4.5	Vlastnosti EAM metody	23
4.6	Výpočet na GPU	24
5	Výpočetní systém Lammmps	26
5.1	Použitý program	26
5.2	Instalace Lammmps	27
5.3	Vstupní soubor Lammmps	29
6	Simulace jednoosého tahu fcc krystalu	38
6.1	Tah elementární buňky při různých teplotách	39
6.2	Tah elementární buňky různých kovů	40
6.3	Tah vzorku konečných rozměrů	42

6	Závěr	46
7	Použitá literatura	47
8	Použité symboly a zkratky	48

1 Úvod

Modelování chování těles na úrovni atomů a molekul je složitý problém. V dnešní době, kdy se začínají vyrábět předměty z atomů, jsou velmi intenzivně zkoumány vlastnosti struktur složených z několika málo atomů. K takové práci je zapotřebí vytvořit matematický model popisující chování částic a v neposlední řadě výpočetní prostředky, které pomocí zadaných vstupních parametrů určí chování soustavy. Pokud popisujeme chování atomů tělesa o určité teplotě, jedná se o úlohu molekulární dynamiky (MD).

Vývoj metod molekulární dynamiky začal okolo roku 1960, jedna z prvních prací na dané téma [1] byla publikována roku 1959. V této době ještě nebylo možné efektivní využití těchto metod v praxi kvůli nedostatku výpočetního výkonu. Několik dalších desetiletí trvalo, než se podařilo naprogramovat a vyladit funkční systémy, které by byly použitelné k řešení praktických problémů. Vývoj jednoho z dnes nejpoužívanějších softwarových balíčků Lammmps začal roku 1990 za spolupráce dvou výzkumných ústavů (Sandia a NNLN) a tří firem (Cray, Bristol Myers Squibb a Dupont). Po pěti letech výzkumu se jim podařilo vytvořit programové vybavení Lammmps použitelné k praktickým výpočtům. Od té doby dodnes program Lammmps integroval spoustu dalších funkcí a vylepšení. Spolu s masivním nárůstem výpočetního výkonu počítačů v několika posledních letech začíná být používání atomistických simulací součástí běžného života.

Molekulární dynamikou lze simulovat takřka libovolné problémy na úrovni atomů a molekul. Nejpoužívanější je k simulacím chování pevných a kapalných materiálů. Přitom nezáleží na složitosti molekulární struktury simulované látky. Velmi často jsou simulace prováděny i pro složité organické sloučeniny. V molekulární dynamice se na atomy nahlíží většinou jako na hmotné body, chovající se podle Newtonových zákonů. Integrací pohybových rovnic lze snadno získat funkce pohybu atomů. Všechny vlastnosti částic jsou zahrnuty ve funkcionálu energie odvozeného pro konkrétní simulovaný systém. Většinou jde o párové potenciály získané metodou ab-initio, nebo pomocí experimentálních měření, kdy jsou pak z makroskopických veličin odvozeny vlastnosti atomů.

Atomistické simulace lze využít i k určování teoretické pevnosti monokrystalů. Pokud známe teoretickou pevnost, můžeme zhodnotit, jak kvalitní materiál se nám podařilo vyrobit, nebo určit, jak hodně lze ještě zlepšovat pevnostní vlastnosti vyráběných materiálů. V molekulární dynamice lze navíc sledovat různé dynamické děje, jako je tání materiálu, rozpouštění, difuze a mnoho dalších procesů.

V této práci bylo vyzkoušeno, jakým způsobem je možné pomocí programu Lammmps vypočítat teoretickou pevnost krystalu v tahu. Oproti jiným běžně používaným metodám (ab-initio výpočty) pro určení teoretické pevnosti tento program dovoluje navíc do výpočtu zahrnout vliv teploty.

Mezi další výhody použitého programu patří možnost převodu výstupního souboru pro programy zobrazující polohy atomů v prostoru. Díky tomu lze sledovat atomy v průběhu deformace.

2 Formulace problému a stanovení cílů

Úkolem této práce je seznámit se s problematikou dynamického modelování procesů v krystalech pevných látek. K tomuto účel poslouží programový systém LAMMPS, který bude nutné zkompilovat a optimalizovat pro paralelní běh na víceprocesorovém klastru. Po uvedení do provozu a ověření funkčnosti systému bude provedena simulace jednoosého tahu dokonalého krystalu. Ze získaných hodnot bude určena mez pevnosti krystalové buňky v tahu.

3 Fyzikální principy

3.1 Teorie vazeb

K popisu pohybu atomů je potřeba znát síly, které mezi nimi působí. Pak už je snadné určit dráhy atomů, jejich rychlosti nebo energii. Je všeobecně známé, že mezi částicemi existují čtyři druhy interakcí, silná a slabá jaderná, gravitační a elektromagnetická. Při určování dynamického chování jednotlivých atomů je nejdůležitější elektromagnetická interakce, vlivy gravitační síly nebo jaderných interakcí jsou zanedbatelné.

Elektromagnetickou interakci matematicky popsal Charles Augustine de Coulomb. Ve své práci uvedl, že síla mezi elektricky nabitými částicemi je nepřímo úměrná čtverci vzdáleností mezi náboji, a přímo úměrná jejich velikosti Q_A a Q_B . A její směr závisí na znaménku výsledku. Velikost této síly tedy je

$$F_{AB} \sim \frac{Q_A \cdot Q_B}{r_{AB}^2}.$$

Ve své době ještě nedokázal určit konstantu úměrnosti, ta byla stanovena později a její velikost je $1/4 \cdot \pi \cdot \varepsilon_0$. Takto doplněný tvar Coulombova zákona platí pouze ve vakuu. Pro všechna jiná prostředí je nutné přidat konstantu $1/\varepsilon_r$. Tato konstanta, ε_r se nazývá relativní permeabilita prostředí. Její hodnota je pro vakuum jedna a ve všech ostatních prostředích větší. Jelikož hodnota relativní permeability je ve vzduchu téměř jedna, její velikost se většinou pro výpočty ve vzduchu uvažuje stejná jako pro vakuum. Coulombův zákon se dnes používán ve tvaru

$$\vec{F}_{AB} = \frac{1}{4 \cdot \pi \cdot \varepsilon_0 \cdot \varepsilon_r} \cdot \frac{Q_A \cdot Q_B}{R_{AB}^3} \cdot \vec{R}_{AB}. \quad (1)$$

Z Třetího Newtonova zákona (akce-reakce) je patrné, že síla působící na náboj B od A bude stejně velká, opačně orientovaná, což je vidět i z rovnice (1). Výsledek bude stejný až na opačné znaménko, které zapříčiní vektor \vec{R}_{AB} .

S použitím Coulombova zákona (1) lze snadno napsat rovnici pro silové působení dvou bodových nábojů na třetí.

$$\vec{F}_B = \frac{Q_B}{4 \cdot \pi \cdot \varepsilon_0} \left(Q_A \frac{\vec{R}_{AB}}{R_{AB}^3} + Q_C \frac{\vec{R}_{CB}}{R_{CB}^3} \right).$$

3.2 Elektrické pole

Abychom nemuseli uvažovat při všech výpočtech polohy všech okolních částic, stačí znát intenzitu a směr elektrického pole v místě, kde leží elektricky nabitá částice, pro kterou chceme počítat velikost síly na ni působící. Že je takovýto postup správný je vidět z mírně upravené rovnice (1)

$$\vec{F} = \left(\frac{1}{4 \cdot \pi \cdot \epsilon_0} \cdot \frac{Q}{r^3} \cdot \vec{r} \right) \cdot q,$$

kde \vec{r} je vzdálenost bodového náboje q od Q . Výraz v závorce závisí pouze na velikosti bodového náboje a vzdálenosti od něj. Tedy okolo bodového náboje vznikne elektrické pole. Síla zmiňovaného náboje umístěná do tohoto pole má velikost

$$\vec{F} = q \cdot \vec{E}(\vec{r}).$$

Elektrické pole, stejně jako jiná vektorová pole, se znázorňuje šipkami. Ty začínají v kladném náboji a směřují k zápornému.

3.3 Práce

Velmi časté je popisovat částice prací, kterou bylo potřeba vynaložit, aby se dostaly do určité polohy. Je zřejmé, že při pohybu elektrického náboje v nehomogenním elektrickém poli se koná práce. Ta vyjadřuje velikost přenesené energie. Za kladnou obvykle považujeme práci, která byla vykonána na sledovaném systému. Pokud tedy systém přijímá energii, konaná práce je kladná. Práce se počítá jako skalární součin vektoru síly a posunutí

$$W_{na} = \int \vec{F} \cdot d\vec{l}. \quad (2)$$

Pokud chceme spočítat velikost práce vykonané přenesením náboje q z jedné do druhé polohy, dosadíme ze vztahu (1) do (2) a dostaneme

$$W_{na} = \frac{1}{4 \cdot \pi \cdot \epsilon_0} \cdot Q \cdot q \cdot \left(\frac{1}{r_2} - \frac{1}{r_1} \right),$$

kde r_1 je počáteční a r_2 koncová poloha náboje q .

3.4 Potenciální energie

Při výpočtech v molekulární dynamice se nejčastěji setkáme právě s potenciální energií. Díky ní lze popsat silovou interakci mezi libovolnými částicemi. Jelikož jde o velmi často používanou veličinu, někdy se výraz potenciální energie nahrazuje slovem potenciál, což je poněkud matoucí, neboť potenciál je označení pro jinou fyzikální veličinu (většinou potenciální energii v rovnovážné poloze). Nulová potenciální energie byla určena pro částici nacházející se v nekonečné vzdálenosti. Potenciální energie ve vzdálenosti r od Q je tedy práce, kterou bychom vykonali přenesením náboje q z nekonečna do vzdálenosti r od Q ,

$$U = W_{na} = \frac{1}{4 \cdot \pi \cdot \epsilon_0} \cdot \frac{Q \cdot q}{r}.$$

Většinou je potřeba určit sílu ze známé potenciální energie. Vztah mezi potenciální energií a silou stanovíme pomocí několika jednoduchých kroků. Celková energie tělesa ε zůstává konstantní a je součtem kinetické a potenciální energie

$$\varepsilon = \frac{1}{2} \cdot m \cdot v^2 + U. \quad (3)$$

Rovnici (3) derivujeme podle času

$$\frac{d\varepsilon}{dt} = m \cdot v \cdot \frac{dv}{dt} + \frac{dU}{dt}.$$

Rychlost je derivací polohy a zrychlení derivací rychlosti, což lze dále upravit na

$$0 = \left(m \cdot \frac{d^2x}{dt^2} + \frac{dU}{dx} \right) \cdot \frac{dx}{dt}.$$

Podle Druhého Newtonova zákona pak pro sílu, která se rovná součinu hmotnosti a zrychlení vyjde rovnice

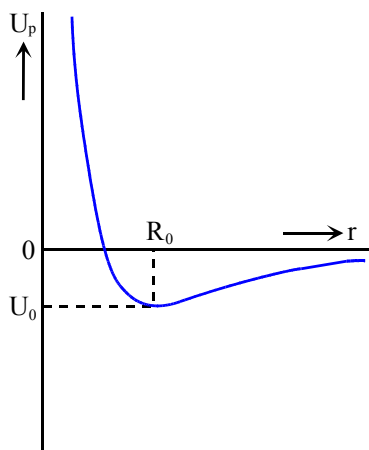
$$F = -\frac{dU}{dx}.$$

Pro tři dimenze analogicky platí

$$\vec{F} = -\left(\frac{\partial U}{\partial x} \vec{e}_x + \frac{\partial U}{\partial y} \vec{e}_y + \frac{\partial U}{\partial z} \vec{e}_z \right) = -\vec{grad}(U),$$

kde \vec{e} jsou jednotkové vektory v osách pravoúhlého souřadnicového systému.

Síly působící mezi částicemi mohou být přitažlivé nebo odpuzivé. Rovnováha mezi těmito silami určuje vzdálenost mezi atomy. Kohezní energie krystalu závisí na atomech (mřížkové konstanty) a na úhlech v buňce. Zjednodušeně se uvádí pouze závislost na vzdálenosti dvou atomů. Schematické vyjádření průběhu potenciální funkce závislé na vzdálenosti dvou atomů je uvedeno na obrázku 1.



Obrázek 1: Graf průběhu potenciální energie v závislosti na vzdálenosti dvou atomů

Jak je patrné z obrázku 1, pro potenciální energii platí, když $r \rightarrow \infty$, $U_p \rightarrow 0$. Velikost potenciální energie dvojice atomů v rovnovážné vzdálenosti odpovídá hodnotě U_0 (vyznačené na obrázku 1). Hodnoty energie U_0 i rovnovážná vzdálenost R_0 závisí na mnoha vlastnostech částic, na elektronovém obalu, na iontovém charakteru vazby a mnoha dalších. Ve většině případů se zanedbává úhlová závislost potenciální energie a uvažuje se pouze centrální párový potenciál. Pak kohezní energii nekonečně velkého krystalu, přepočtenou na jeden atom, lze vyjádřit ve tvaru

$$U(r) = \frac{1}{2} \sum_{i=1}^{\infty} U_{pi}(r_i),$$

kde r_i je vzdálenost od libovolného referenčního atomu a sumace se provádí přes všechny ostatní atomy v krystalu. Znalost funkce $U(r)$ umožňuje výpočet mechanických charakteristik, jako jsou ideální pevnost nebo elastické koeficienty.

3.5 Vazby v pevných látkách

Molekulární dynamika umí řešit prakticky jakýkoliv druh problému, třeba i proudění částic plynu nebo kapaliny. V případě problémů v mechatronice se nejčastěji používá ke studiu pevných látek. Abychom mohli sledovat jejich chování, je potřeba získat párové potenciály, které popisují silové působení dvou částic

v závislosti na jejich vzdálenosti. Při určování těchto potenciálů jsou nejdůležitější kohezní síly, které „drží“ pevné látky pohromadě. Tyto síly způsobuje elektrostatická interakce mezi zápornými elektrony a kladnými jádry atomů. Existují rozdíly mezi typy vazeb dané různým rozložením vnějších elektronů a iontových zbytků. Rozlišujeme iontovou, kovalentní, kovovou, van der Waalsovu a vodíkovou vazbu.

Vazby v pevných látkách byly rozděleny podle vlastností pro ně charakteristických do několika skupin. Iontová vazba je určitý druh polární vazby, u které rozdíl elektronegativit atomů přesahuje hodnotu 1,7. Kovalentní vazba je chemickou vazbou, která je charakteristická sdílením jednoho nebo více elektronových párů z valenčních vrstev elektronového obalu mezi dvěma atomy. Vodíková vazba, někdy nazývaná vodíkový můstek, je druhem slabé vazebné interakce mezi molekulami nebo v rámci dvou částí jedné molekuly. Je podstatně silnější, než jiné mezimolekulární vazby, ale o řád slabší, než iontová nebo kovalentní vazba. Vodíková vazba vzniká mezi atomem vodíku a nevazebným elektronovým párem silně elektronegativního atomu, který odsaje elektron od kladně nabitého jádra vodíku, ten pak může poutat nevazebné elektronové páry okolních molekul. Velmi slabé síly, které se výrazně projevují pouze u atomů vzácných plynů, jsou důsledkem Van der Waalsovy vazby. Ta samozřejmě existuje současně vedle iontových a kovalentních vazeb, ale vzhledem k tomu, že je o mnoho řádů menší, uvažujeme ji pouze u vzácných plynů. A konečně vazba charakteristická pro nejčastěji používané materiály je vazba kovová. U kovové vazby jsou valenční elektrony sdílené všemi jádry. Tento elektronový „plyn“ způsobuje dobrou tepelnou a elektrickou vodivost kovů.

V jakékoliv pevné látce jsou účinné výše zmíněné chemické vazby a nezáleží, jedná-li se o amorfni nebo krystalickou látku. Veličinou charakterizující stabilitu vazeb mezi částicemi je kohezní energie U , lze ji chápat jako energii uvolněnou při vzniku krystalu. Tedy rozdíl celkové energie volných atomů E_v a celkové energie krystalu E , složeného ze stejného počtu atomů ($U = E_v - E$). Vyšší kohezní energie znamená vyšší stabilitu krystalu. Obecně je za nejstabilnější formaci atomů považován monokrystal. U takového útvaru lze očekávat nejvyšší kohezní energii.

3.6 Kovová vazba

Pro kovy je charakteristická vysoká pohyblivost valenčních (vodivostních) elektronů a právě tyto elektrony jsou velmi významné při určování vazební energie. Tento typ vazby si lze představit jako soustavu kladných nábojů (jader) ponořenou v téměř homogenním plynu vodivostních elektronů. Kohezní energie kovů se odvozuje pomocí Wignerovy-Seitzovy metody (více o metodě je uvedeno v knize [2]). Princip spočívá v tom, že vytvoříme Wignerovu–Seitzovu buňku v oblasti kolem

bodů (atomů) kovové mřížky, pro jejíž body platí, že jejich vzdálenost k atomu, kolem kterého je vytvořena je menší, než k jakémukoliv jinému uzlovému bodu dané mřížky. U kovů lze každou Wignerovu-Seitzovu buňku aproximovat koulí o poloměru r , ta bude představovat jeden kationt. Celková potenciální energie kovu je obecně dána interakcí nábojů uvnitř jednoho mnohostěnu a interakcí mnohostěnů. Takto vytvořené mnohostěny budou navenek elektricky neutrální a interakci mezi nimi lze zanedbat, aniž bychom se dopustili výraznějších chyb. Celková energie krystalu je dána součtem kinetické a potenciální energie všech elektronů. Kinetickou energii elektronů můžeme určit tak, že valenční elektron budeme považovat za elektron volný a užijeme Fermiho statistiku. Fermiho energie částic je energie nejvyššího obsazeného stavu. Jednoduše řečeno, pokud bychom materiál ochladili na teplotu absolutní nuly, elektrony by se stále pohybovaly – ten nejrychlejší by měl kinetickou energii stejně velkou, jako je právě Fermiho energie. Podle této teorie je střední energie systému volných elektronů $3/5 E_F$ (E_F je Fermiho energie) a pro kinetickou energii platí

$$E_k = \frac{3}{5} E_F = \frac{3}{10} \frac{h^2}{m r^2} \left(\frac{9\pi}{4} \right)^{\frac{2}{3}}.$$

V uvedeném vztahu je m hmotnost elektronu, $h = 2\pi\hbar$ je Planckova konstanta. Spočítat potenciální funkci je o něco složitější, v některých případech se dají použít experimentálně naměřené hodnoty. Minimum součtové funkce potenciální a kinetické energie určuje rovnovážnou vzdálenost (mřížkový parametr). Z těchto výsledků je možno získat kohezní energii počítanou vzhledem k soustavě volných atomů nekonečně od sebe vzdálených. Označíme E_I ionizační energii volného atomu, a E_0 potenciální energie elektronů v poli kladného iontu, pak na jeden atom připadá kohezní energie

$$U = - \left[E_0 + \frac{3}{5} E_F + E_I \right].$$

Kohezní energie je vždy kladná veličina, ve vztahu je záporná pouze hodnota E_0 .

3.7 Vliv teploty

Existuje víc způsobů jak počítat s teplotou, jedním z nejpoužívanějších je zahrnutí teploty do potenciální energie. Tento postup lze snadno ukázat na dvou dipólových atomech. Vzdálenost dvou atomů zůstane konstantní, ale jejich úhly natočení se budou měnit. Některé orientace dipólů se stanou energeticky více výhodnými, což zohledníme

zahrnutím Boltzmannovy konstanty k_b . Aby bylo zřejmé, že se jedná o velikost střední hodnoty, použijí se závorky $\langle \dots \rangle$. Tepelná potenciální energie dvou molekul je

$$\langle U_{AB} \rangle = \frac{\int U_{AB} \exp\left(-\frac{U_{AB}}{k_B T}\right) d\tau}{\int \exp\left(-\frac{U_{AB}}{k_B T}\right) d\tau},$$

kde T je termodynamická teplota. Výsledná teplota se spočítá integrací přes všechny možné úhly natočení τ , přičemž vzdálenost molekul zůstává konstantní.

Pro výpočty v molekulární dynamice je daleko výhodnější počítat s teplotou jako s kinetickou energií, než měnit potenciálové funkce. Pokud zahrneme teplotu do kinetické energie atomů, je potřeba volit velmi malý integrační krok pro výpočet polohy (obvykle tisíciný pikosekundy). Kinetická energie se pak počítá podle vztahu

$$\frac{1}{2}mv^2 = \frac{\text{stupně volnosti}}{2Nk_bT},$$

kde N je počet atomů, k_b Boltzmannova konstanta a T termodynamická teplota, přičemž stupně volnosti jsou pouze translace (rotace pro body nemají smysl).

3.8 Odpudivé síly

Tyto síly nejsou podstatné pro vznik krystalu, ale spoluurčují rovnovážnou vzdálenost. Působí při velmi malých vzdálenostech jader atomů a jejich příčinou je odpudivá síla jader a překrývání elektronových slupek. Uvnitř atomu převažuje elektrostatické pole kladného náboje a při proniknutí jádra jiného atomu dovnitř působí tedy vždy odpudivá síla. Nejsou přesně známy všechny vlivy podílející se na jejím vzniku, ale z experimentů je zřejmé, že tato síla se zmenšující se vzdáleností jader velice prudce roste. Její tvar nejlépe vystihuje exponenciála. Odpudivá síla se obvykle popisuje vztahem

$$U_{rep} = Ae^{-Br},$$

kde A a B jsou materiálové konstanty, r je vzdálenost dvou atomů.

4 Metoda vnořeného atomu a postup výpočtu

4.1 Hlavní problémy

Pomalu ve vývoji procesorů dochází k situaci, kdy už není technicky možné výrazně zvyšovat výpočetní výkon zvýšením pracovní frekvence. Pro současný stav je charakteristická integrace mnoha jader do jednoho procesoru. Dnes vyráběné počítače mají relativně pomalé procesory, dva až tři GHz, avšak obsahují dvě nebo větší počet jader. Ve stejný okamžik je takový počítač schopen provádět více početních operací najednou. Aby bylo možné využít naplno výpočetní výkon takové sestavy, musí se provést paralelizace algoritmů. Všechny náročnější výpočty, ke kterým patří i výpočty v molekulární dynamice je tedy nutné upravit tak, aby mohly využívat vícejádrové procesory.

Výpočet molekulární dynamiky spočívá v nalezení polohy atomu s minimální potenciální energií, v rámci možnosti pohybů atomů. Lze si představit problém tak, jako kdyby byly atomy pospojované pružinkami, začali jsme na některé z nich působit silou, nechali systém relaxovat a poté hledali polohy, ve kterých se atomy ustálí. Existují i systémy pro výpočty molekulární dynamiky založené na modelu kuliček pospojovaných pružinami, nicméně většinou se už nepoužívají.

Sílu, kterou potřebujeme znát, abychom mohli určit chování atomu, nemusíme vždy počítat u všech částic. Potenciály mají pochopitelně nekonečný dosah, ale po určité vzdálenosti je velikost potenciální energie pro danou dvojici atomů zanedbatelná. Známe velké množství různých potenciálů, pro výpočty v kovových krystalech se nejčastěji používá potenciál vnořeného atomu. Konečný dosah potenciálů je velmi důležitá vlastnost, díky které je možné provést efektivní paralelizaci pro tisíce procesorů.

Další otázkou je, jak jednotlivé atomy přiřadit různým procesorům, aby byl výpočet co nejefektivnější. Dvě nejpoužívanější metody jsou rozděleny podle působících vnějších sil, nebo podle geometrie systému.

Pokud geometricky oddělíme simulovanou oblast a procesoru přiřadíme atomy ležící v objemu o určitých souřadnicích, může se stát, že všechny atomy se během simulace z daného objemu přesunou pryč, takový procesor by zůstal dále nevyužitý. Pokud naopak přidělíme procesoru skupinu atomů, které spolu sousedí, během simulace se mohou promíchat a jelikož pro výpočet síly je nutné znát polohy blízkých atomů, pak pokud tyto atomy náleží různým procesorům, je výpočet značně neefektivní. Nedá se říct, že některý z použitých postupů výpočtu by byl nejlepší pro všechny druhy problémů. Pro některé simulace je lepší použít rozdělení atomů procesorům na základě působících sil, pro jiné podle jejich polohy v prostoru.

4.2 Použití

Metoda vnořeného atomu není pochopitelně vhodná pro všechny typy simulací. Například řešení chování směsí více druhů molekul (jakou je třeba voda) je příhodnější použít potenciály velkých dosahů, jakými jsou například Lenardovy-Jonesovy potenciály (v Lammps jsou integrovány v Ewaldově algoritmu). Naproti tomu takřka ideální je pro výpočty chování kovových mřížek pevných krystalů, jako jsou simulace růstu trhlin, vznik reliéfů a podobně. Při výpočtech chování kovových materiálů vyniká metoda EAM mezi ostatními zejména svoji rychlostí, ale i přesností. EAM potenciály nemají příliš velký dosah, tudíž je výpočet rychlejší a snadněji paralelizovatelný.

Je zřejmé, že nelze dosáhnout toho, aby doba výpočtu byla nepřímo úměrná počtu procesorů, nicméně metoda EAM se takovému vztahu, zejména pro větší počty atomů, úspěšně přibližuje. Dalším kritériem, které metoda EAM splňuje, je aplikovatelnost na libovolnou geometrii při zachování dostatečné efektivity výpočtu.

4.3 Princip výpočtu

Aby bylo možné dojít po určité době k výsledkům, nelze v našem případě počítat podle analyticky přesných Schrödingerových rovnic. Toto řešení lze použít pouze pro některé vhodné situace-například výpočty s atomy vodíku, pravidelné mřížky a podobně. Proto se pro výpočty molekulární dynamiky (MD) používají vztahy z mechaniky pro hmotné body (Newtonovy zákony). Energii i -tého atomu popisuje rovnice

$$E_i = \sum_j \phi(r_{ij}) + F_v \left(\sum_j \rho(r_{ij}) \right), \quad (4)$$

kde ϕ je interakce párového potenciálu mezi atomy i a j a F je vnořující funkce, konkrétně energie vnořeného atomu i v elektronové hustotě ρ poskytnuté j atomy, které ho obklopují [3]. Hodnoty funkcí ϕ , ρ , a F jsou analyticky vyjádřené s koeficienty odvozenými na základě mnoha různých experimentů. Všechny interakce mají konečný dosah. Tedy všechny sumace v rovnici (4) se počítají přes několik málo atomů j , které obklopují atom i . Abychom získali sílu \vec{F} působící na atom i , derivujeme rovnici (4)

$$\vec{F} = - \sum_j \left[\phi'_{ij} + F'_i \rho'_{ij} + F'_j \rho'_{ji} \right] \vec{r}_{ji}, \quad (5)$$

kde F'_i je derivací i -té vnořené funkce F_v a ρ'_{ij} je derivací elektronové hustoty atomu j v místě i -tého atomu. Pro výpočet je nutné určit velikost F' každého j -tého souseda atomu i . To znamená, že v EAM jsou síly počítané ve dvou krocích. Nejprve jsou

v rovnici (4) sečteny vnořující funkce (a současně s tím vyčísleny jejich derivace). Tento postup se provede pro všechny atomy. Pak je spočítaná rovnice (5) s použitím hodnot F' . Aby bylo možné počítat tyto rovnice současně, je potřeba umožnit komunikace mezi procesory během výše popsaných dvou kroků výpočtu. V prvním kroku procesor, který vlastní atom i zjistí jemu náležící hodnoty F a F' . Poté odešle hodnotu F' všem jeho j sousedům (procesory vlastnící okolní atomy) obklopujícím tento atom, pak mohou být pro každý atom určeny rovnice (5). Dále je potřeba věnovat úsilí zjednodušení (zefektivnění) komunikace mezi jednotlivými procesory, ta je totiž velmi pomalá.

4.4 Paralelní algoritmy

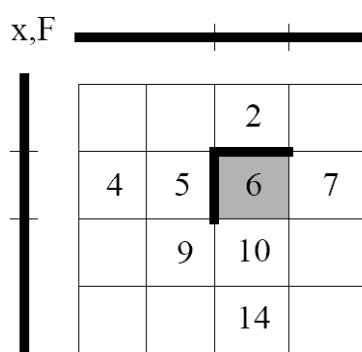
Pokud do simulace běžící na P procesorech vstupuje N atomů, je intuitivním postupem přiřadit N/P atomů každému procesoru na celou dobu výpočtu. Tento přístup se označuje jako „atomová dekompozice“ pracovní oblasti. Každý procesor počítá síly působící pouze na své atomy a následně aktualizuje jejich pozice. Při výpočtu síly potřebuje každý procesor znát polohy okolních atomů, aby mohl zjistit funkce ϕ a ρ v rovnicích (4) a (5). Proto musí během každého výpočetního kroku (jeden krok = jedna integrační iterace) poslat každý procesor všem ostatním informace o polohách jemu přiřazených atomů. Stejně tak při výpočtu síly (rovnice 5) je potřeba získávat informace o okolních atomech přiřazených různým procesorům, tudíž je potřeba navíc další čas k výměně dat.

Komunikace, během které jeden procesor rozešle informace všem ostatním procesorům a zároveň od nich informace získá je provedena v $\log_2(P)$ krocích. Čas trvání přenosu dat je očividně závislý na celkovém množství informací, které musí být přeneseny. Počet kroků potřebných k přenosu informací mezi procesory je závislý na počtu procesorů. Pokud je jich méně než 64, není tato operace příliš časově náročnou. V opačném případě, za vyššího počtu procesorů, stává se tato část výpočtu časově nejnáročnější. Pro malé počty N a při omezeném množství procesorů je zmiňovaná metoda dostatečně efektivní z hlediska výpočetní rychlosti i náročnosti na naprogramování.

Metodu popsanou výše je pro složitější výpočty potřeba upravit, aby se potlačila zejména její nejslabší část – výměna dat mezi procesory. Místo náhodného přiřazení atomů procesorům se provede přiřazení podle umístění atomů v prostoru. Každému procesoru je v tomto případě přidělena určitá část simulované oblasti, tzv. pod-blok. Takový postup se odborně nazývá „prostorová dekompozice“ pracovního prostoru. Každý procesor počítá síly působící na atomy v jeho oblasti s tím, že jak se pohybují, přijímá během výpočtu nové, které do něj vstupují, a předává atomy, které vystupují. Jelikož silové interakce mezi příliš vzdálenými částicemi nejsou podstatné, procesoru stačí komunikovat pouze s několika málo procesory, které „vlastní“ atomy v blízkém okolí

atomů, jejichž polohy počítá. Samozřejmě k tomuto přibývá nutnost výměny informace o atomu, který přechází z jednoho procesoru do druhého. Tento algoritmus je vhodný zejména pro velký počet atomů. Množství informací, které se vyměňují mezi procesory, odpovídá poměru povrch/objem oblasti zpracovávané určitým procesorem. To lze v trojrozměrné oblasti vyjádřit vztahem $(N/P)^{2/3}$. Rozhodnout se pro užití zmiňovaného výpočetního postupu lze na základě poměru N/P . Pokud bude vycházet příliš malé číslo, byla by tato metoda neefektivní; dosah potenciálů by byl velký vůči velikosti geometrické oblasti jednoho procesoru. Dále je potřeba zajistit, aby se simulovaná oblast skládala z pravidelných bloků obsahujících stejný počet atomů - aby výpočetní zátěž byla rovnoměrně rozdělena mezi jednotlivé procesory. Toho se dosahuje velmi obtížně, zejména pokud atomy nevyplňují simulovanou doménu pravidelně, nebo se nerovnoměrně přesunují během výpočtu. Například pokud působí někde na povrch síla, mnoho procesorů zůstane nevytížených kvůli tomu, že se atomy z jejich oblastí (kolem působíště síly) přesunou pryč.

Problém „vyprazdňování“ domén řeší algoritmus, který se jmenuje „silová dekompozice“ pracovního prostoru. Tento postup se snaží přiřadit atomy jednotlivým procesorům podle působících sil. Tím se zamezí tomu, aby přecházely mezi doménami, kterým jsou přiřazené. Doba výpočtu zhruba odpovídá poměru N/\sqrt{P} , což je určité zlepšení oproti atomové dekompozici, avšak metody prostorové dekompozice dosahují lepších výkonů (N/P). Tyto odvozené vztahy nezohledňují různé další vlivy, jako jsou malé propustnosti sběrnic a podobné, takže se ukazuje, že metoda silové dekompozice je obvykle pro simulaci 10 000 tisíc atomů oproti očekávání rychlejší, než metoda prostorové dekompozice a pro případy se 100 000 atomy už jsou rozdíly v rychlosti obou metod nepodstatné.



Obrázek 2: Rozdělení úloh pro procesory v $N \times N$ matici sil v algoritmu silové dekompozice. Každé políčko označuje jeden procesor: jedna souřadnice je F (vektor sil) a druhá x (vektor poloh). Procesor 6 musí uchovávat informace o polohách a působících silách každou o velikost N/\sqrt{P} . Převzato z [4]

Stejně jako u metody atomové dekompozice, i v metodě silové dekompozice je každému procesoru přiřazen pevný počet atomů (N/P) po celou dobu běhu simulace. Výpočetní operace, které jsou potřebné k výpočtu sil působících na atomy lze rozepsat do „silové“ matice $N \times N$. Každý element této matice, který má šířku $i-j$ (y -ová osa na obrázku 2) popisuje silové interakce mezi atomy i a j . Celková síla působící na atom i je tedy součtem všech elementů v řádku této matice. Takto vytvořená matice je rozptýlená (popisuje síly s krátkým dosahem) a symetrická (podle Newtonova třetího zákona). Všechny procesory obsahují jeden prvek silové matice, jak je vidět z obrázku 2. Toto je určitá změna, oproti prostorově dekompoziční metodě, podle které by každému procesoru připadlo N/P řádků takovéto matice. Pokud jsou atomy uspořádány náhodně, každý blok bude mít zhruba stejný počet nenulových elementů. Pro výpočty elementů matice patřící jednotlivým procesorům nepotřebujeme znát pozici všech N atomů, ale pouze polohy dvou podskupin atomů, každá o velikosti N/\sqrt{P} (viz obr 2).

Pro výpočet je nutná komunikace mezi malými skupinami \sqrt{P} procesorů, které vyžadují výměnu N/\sqrt{P} informací. Zde je patrné zlepšení oproti atomové dekompozici, kde všech P procesorů si přeměňuje N informací. Celkem tedy dochází k transferu o \sqrt{P} méně informací mezi procesory, než u dekompoziční metody, a o \sqrt{P} více, než vyžaduje prostorová dekompozice.

Metodu silové dekompozice lze dále vylepšit tak, že využijeme 3. Newtonova zákona, čímž se sníží výpočetní kroky na polovinu. Tato operace se provádí pomocí řízeného postupu silovou maticí. Elementy nad maticovou diagonálou jsou 0, pokud $(i+j)$ je sudé a elementy pod diagonálou jsou nulové, pokud $(i+j)$ je liché. Každá párová interakce je v matici počítána pouze jednou. Pak ale vyvstává nutnost poslat informaci o zjištěné síle okolním procesorům ještě předtím, než se provede řádková sumace.

4.5 vlastnosti EAM metody

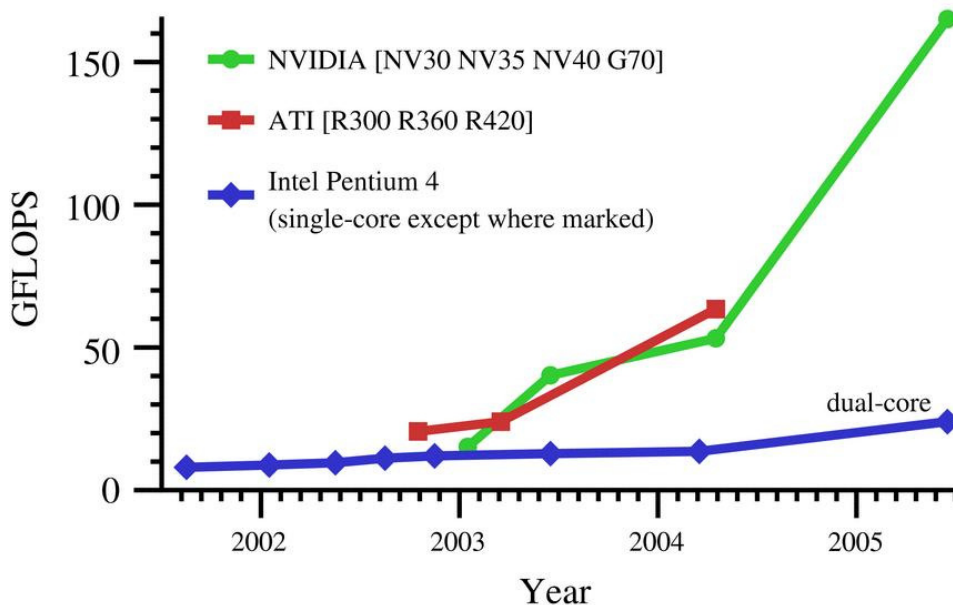
V předchozí části byly rozebrané dva hlavní principy algoritmizace EAM metod. Jejich společnou předností je, že nepotřebují znát informace o geometrickém (prostorovém) uspořádání simulované oblasti. To je jeden z důvodů, proč lze jednoduše rozdělit výpočet rovnoměrně mezi procesory. Obě metody se dají relativně jednoduše naprogramovat. Metoda silového rozkladu dosahuje efektivnosti při paralelizaci (50 %) při použití u rozsáhlých problémů. Umožňuje nám simulovat celky atomů malých i velkých rozsahů během zlomku času, které by tyto úlohy zabraly na konvenčních vektorových superpočítačích.

I když se zabýváme paralelizací metod počítajících potenciály EAM, tento postup je aplikovatelný i na jiné potenciály používané v MD. Paralelizace vycházející z výše uvedených principů mohou být aplikované i na jiné (například L-J) párové potenciály. Metoda prostorové dekompozice dobře pracuje i s tříčasticovými potenciály (např. pro Si a oxid křemičitý, různé slitiny kovů) [5] nebo i pro metody vyšších řádů jako jsou

mnohočasticové výpočty s fonony a zkoumání defektů způsobených radiací [6]. Nedávno se také podařilo rozšířit metodu silové dekompozice na kovalentně pospojované systémy, mezi které patří zejména polymery [7]. Pro takovéto skupiny je navíc potřeba spočítat 2, 3 a 4 částicovou (vazebnou) interakci. Tyto algoritmy lze velice jednoduše naprogramovat pro libovolné paralelní systémy s distribuovanou pamětí, které zvládají výměnu zpráv mezi procesory (MIMD paralelní počítač).

4.6 Výpočet na GPU

Během posledního roku se objevila nová možnost využít grafické karty k výpočtům používaným v molekulární dynamice. (GPU = grafická výpočetní jednotka) Společnost Nvidia představila nové jádro G80, které je plně programovatelné na rozdíl od předchozích verzí, u kterých byl omezený počet a typ instrukcí, které uměly vykonávat. Grafické čipy obsahují paralelní procesory s několika jádry. Ty jsou srovnatelné se superpočítači, což je několik stovek počítačů zapojených v síti a dělících si výpočetní úkoly mezi sebou. Grafické čipy nepřinášejí něco nového - paralelizace se používá při výpočtech už dlouhou dobu, podstatný rozdíl spočívá v ceně. Grafická karta stojí sto až tisíckrát méně, než superpočítač se stejnými výpočetními schopnostmi. Srovnání výpočetního výkonu nejběžnějších typů procesorů je na obrázku 3



Obrázek 3: Graf výpočetního výkonu různých typů procesorů
Převzato z adresy [8]

Dnes je pro výpočty v molekulární dynamice nejpoužívanější grafická karta Nvidia s firemním jménem Tesla osazená procesorem C870, který obsahuje 128 paralelních výpočetních jednotek; paměť integrovaná na kartě je 1.5GB s propustností 76.8 GB/s (pro „burst“ mode, při náhodném přístupu je přenos pomalejší) [9]. Výpočetní výkon této grafické karty je 518 gigaflops. Dalšího zvýšení výkonu se dosahuje osazením jednoho počítače více takovými kartami, čímž lze s běžným stolním počítačem dosáhnout výpočetního výkonu až 8 teraflops [10]. Flops je zkratka pro počet operací v plovoucí řádové čárce za sekundu, který se velmi často používá pro srovnávání výpočetního výkonu (z anglického Floating-point operations per second [11]).

Skupinou zabývající se výpočty molekulární dynamiky na grafickém procesoru je Theoretical and Computational Biophysics group [9]. Zaměření této skupiny je primárně spjaté s výzkumem v oblasti farmacie. Nicméně jimi vyvinutý NAMD Molecular dynamics Simulator naprogramovaný speciálně pro výpočty na grafické kartě by měl být bez problémů použitelný i pro výzkum materiálů ve strojírenství.

Vyvíjet programy přímo pro daný hardware není příliš jednoduché. Je zapotřebí znát architekturu procesoru a příkazy, které umí vykonávat (assembler). Aby vývoj programů pro grafické procesory nebyl tak náročný, vytvořila Nvidia balík nástrojů CUDA, který obsahuje vývojové prostředí v jazyce C obsahující Překladač kódu C pro procesory Nvidia, knihovny FFT a BLAS pro GPU, gdb debugger pro GPU a mnoho dalšího pro zjednodušení tvorby programů pro grafické čipy [12].

5 Výpočetní systém lammps

5.1 Použitý program

Celý název programu je: LAMMPS Molecular Dynamics Simulator. Slovo Lammps je nejspíš odvozeno (jak naznačují autoři programu) ze slova lamp, což znamená přístroj generující světlo, teplo nebo terapeutickou radiaci; jiný význam pak: něco co osvětluje duši nebo myšlení [13]. Programový kód lze přeložit pro sériové výpočty (jeden procesor) nebo paralelní, přičemž pro paralelizaci se využije prostorová dekompozice simulované domény. Tento program je velice intenzivně vyvíjen, každým okamžikem přibývají nové funkce a optimalizace.

Program Lammps lze provozovat na jednom nebo více procesorech. Paralelizace je založena na systému distribuované paměti. Simulovaná oblast je geometricky rozdělena na části, které pak zpracovávají jednotlivé procesory. Celá distribuce Lammps je šířena pod licencí open-source software, což znamená, že při dodržení jistých podmínek lze program zcela zdarma užívat, prohlížet a upravovat. Program je napsaný v jazyce C++, což zajišťuje jeho přenositelnost, na libovolnou platformu, bohužel tím také přibývá nutnost program před spuštěním přeložit. Lze simulovat atomy, polymery, molekuly, kovy, bodové dipóly, elipsoidy a jejich kombinace. Mezi použitelné potenciály v Lammps patří například párové (Lennard-Jonsovy, Buckinghamovy, Morseovy), potenciály nabitých částic (Coulombické, bodových dipólů), mnohočásticové (EAM, modifikované EAM (MEAM), Stillinger-Weberovy, Tersoffovy, AI-REBO), hrubozrnné potenciály (DPD, REsquared, coloidní), vazbové potenciály (harmonický, FENE, Morse, nelineární, quartikové (přerušitelné)), úhlové potenciály (CHARMM, cosine), klínové potenciály (harmonické, CHARMM, multi-harmonické, OPLS), nepravé potenciály (harmonické, cvff), hybridní potenciály (multiple párové, úhlové), přetěžované potenciály (superpozice více potenciálů), potenciály polymerů (all-atom, united-atom), potenciály vody (TIP3P, TIP4P, SPC), potenciály implicitních rozpouštědel (hydrodynamické zvlhčovačlo, Debye). Simulovat lze buď trojrozměrné problémy, nebo i dvojrozměrné. Souřadnice takových systémů mohou být pravoúhlé nebo i kosoúhlé (triklinické systémy). Jelikož jde o program molekulární dynamiky, lze měnit teplotu a tlak v simulované oblasti. Simulovaná oblast vyplněná molekulami se dá deformit tlakem nebo smykem. V programu je předdefinováno mnoho různých druhů hranic atomů. Kromě rovnovážných molekulárních simulací lze provést i nerovnovážné simulace non-equilibrium molecular dynamics (NEMD).

5.2 Instalace Lammeps

Program Lammeps může být provozován na mnoha platformách. Dodává se jako zdrojový kód a každý si ho přeloží podle počítačového vybavení, které má k dispozici. Oficiálně podporovaný operační systém je Linux, ale některým lidem se podařilo přeložit a spustit Lammeps i ve Windows.

Výpočty s velkým počtem atomů (desítky milionů) jsou pro osobní počítače časově dost náročné, proto se provádějí na výkonnějších výpočetních serverech. Doporučený protokol pro komunikaci s Linux servery z prostředí Windows je ssh. Většina programů, potřebných pro běh programu Lammeps, nejsou standardně v distribuci Linuxu. Program pro přeložení a kompilaci Lammeps je většinou nutné doinstalovat. Pro přeložení lze použít program gcc stáhnutelný z repozitářů. Jelikož balík Lammeps neobsahuje příliš kvalitní postprocesor, je dobré použít nějaký specializovaný program na zobrazování atomů, například AtomEye. Bohužel tato platforma nepodporuje výstupní formát Lammeps (dump.file). Jako konvertor je doporučeno použití programu Pizza, s hotovými procedurami pro převod typů souborů. Další informace při řešení potíží při překladu programu nebo práci v Linux můžete najít v [14]. Pokud není Linux instalován přímo na harddisk počítače, je důležité věnovat dostatečnou pozornost nastavení. Při použití emulátorů operačního systému tzv. “virtual machine” je nutné u procesoru zapnout podporu virtualizace. V biosu možnost device configuration - virtualization technology – enable. Přesto bude výpočetní výkon o něco nižší, než při přímé instalaci.

Použité programy lze stáhnout z následujících umístění:

1, Lammeps

<http://lammeps.sandia.gov/download.html>

2, mpich2 (paralelní knihovna)

<http://www.mcs.anl.gov/research/projects/mpich2/downloads/index.php?s=downloads>

3, knihovna pro výpočty (Fourierova transformace)

<http://www.fftw.org/download.html>

4, pizza

<http://www.cs.sandia.gov/~sjplimp/pizza.html>

5, AtomEye

<http://mt.seas.upenn.edu/Archive/Graphics/A3/A3.html>

6, virtuální server (pro všechny druhy Windows, lze k němu stáhnout již nainstalovaný linux)

<http://www.vmware.com/>

Po instalaci Mpich2 můžeme program přeložit. Úprava hlavičky souboru (ve složce MAKE) *Makefile.jmeno* (při vytváření hlavičkového souboru je výhodné vyjít už z připraveného *Makefile.linux* a pouze změnit/přidat některé parametry).

```
CC =          icc
CCFLAGS =     -O          -DFFT_FFTW          -DLAMMPS_GZIP          -
DMPICH_IGNORE_CXX_SEEK
DEPFLAGS =    -M
LINK =        icc
LINKFLAGS =   -O
USRLIB =      -L/home/lambers/install/fftw-bin/lib -lfftw -lmpi -llam -llammpi++
SYSLIB =      -lpthread -lstdc++
ARCHIVE =     ar
ARFLAGS =     -rc
SIZE =        size
```

Příkazem *make jmeno* by mělo dojít k přeložení programu.

Program se nashoduje pomocí funkce *mpirun*, kde je možné zadat pomocí parametru *np* počet procesů, které se mají spustit. (pokud navíc jsou počítače zapojeny v „kruhu“, lze tímto příkazem spustit procesy i na nich). K ověření funkčnosti programu je vhodné použít některý z připravených vstupních skriptů ve složce *examples* nebo *bench*.

```
mpirun -np2 lmp_jmeno < in.nazev
```

Ve složce, kde byl program spuštěný, by se měl vygenerovat výstupní soubor s názvem *dump.nazev*. Tímto je ověřeno, že program funguje správně, a může se začít používat k výpočtům.

Příkaz *mpirun* podporuje několik parametrů, jak je možné daný soubor spustit, nejužitečnější jsou *-np počet_procesů*, určí kolik procesů se má nashodovat na jednom počítači; *C*, spustí program na všech dostupných procesorech (pro každý počítač se spustí tolik procesů, kolik je definované v *MPI_COMM_WORLD*). Pokud chceme spustit program jen na jednom uzlu, použijeme parametr *n* a hned za ním číslo uzlu.

5.3 Vstupní soubor Lammeps

Vstupní soubor se píše podle několika pravidel, bez jejichž znalosti je obtížné pochopit, co daný skript dělá. Každý příkaz je proveden v okamžiku přečtení ze vstupního souboru, v případě, že změníme nějaký parametr, projeví se až v za ním následujících příkazech. Některé z nich jsou platné pouze, pokud následuje další a naopak některé musí nějaké jiné předcházet.

- ✓ Každý řádek vstupního souboru, který není prázdný, je vstupním příkazem.
- ✓ Program rozlišuje velká a malá písmena. Přitom platí dohoda, že příkazy se píší malými písmeny stejně jako argumenty příkazů. Velká písmena se používají pro názvy souborů a řetězců.
- ✓ Pokud potřebujeme zadat dlouhý příkaz přes více řádků, použijeme na konci prvního řádku symbol "&". Několik takto pospojovaných řádků se chová, jako by byly napsané na jednom.
- ✓ Všechny řetězce začínající "#" jsou považovány za komentář a nevyhodnocují se.
- ✓ Označování proměnných se provádí znakem \$. Lze napsat buď pouze \$jméno_proměnné, nebo jméno proměnné uzavřít do závorek za symbol \$, tedy \${jméno_proměnné}.
- ✓ Každý řádek vstupního souboru obsahuje příkazy a parametry, ty se oddělují mezerami. Neexistuje omezení, udávající, jaké znaky by se měly používat pro názvy příkazů proměnných nebo parametrů.
- ✓ Při vyhodnocování se na začátku každého řádku hledá příkaz, všechny následující informace jsou brány jako argumenty.
- ✓ Text s mezerami ohraničený apostrofy je považován za jeden výraz.

Další informace o příkazech a psaní vstupních souborů jsou uvedeny v manuálu [15].

Vstupní soubor lze rozdělit do několika částí, každá obsahuje pro ni typické příkazy.

Inicializace: Nastavuje parametry, které se později použijí při rozmisťování atomů do simulované oblasti.

Sekce definic: v LAMMPS existuje víc možností, jak definovat strukturu částic. Pro složité struktury obsahující velké množství druhů atomů velmi nepravidelně uspořádaných (polymery a podobně) lze načíst definici struktury ze zvláštního souboru. Jednodušší modely je možné vytvořit přímo ve vstupním souboru příkazy. Molekuly je nutné definovat pomocí jejich topologie, u struktur složených z atomů stačí zadat jejich mřížku. Atomy v uzlových bodech mřížky vytvoří příkazy *lattice*, *region*, *create_box*, *create_atoms*. Celý blok atomů lze navíc duplikovat příkazem *replicate*, což lze s výhodou využít u rozsáhlejších simulací.

Nastavení: Pokud vytvoříme molekulární strukturu, v dalším kroku pro ni musíme určit vlastnosti. Koeficienty silového zatížení se nastavují následujícími příkazy například *pair_coeff*, *bond_coeff*, *angle_coeff*, *dihedral_coeff*. Parametry simulace nastaví příkazy: *neighbor*, *timestep*, *reset_timestep*, *run_style*. Okrajové podmínky, časová integrace a diagnostické nastavení provádí příkaz *fix*. Různé výpočty mohou být specifikované tak, aby se provedly během výpočtu, mezi takové patří například *compute*, *compute_modify*. Zapsání výstupu do souboru (nebo na obrazovku) provedou příkazy *thermo*, *dump*, a *restart*.

Spuštění simulace: Molekulární dynamická simulace se spustí příkazem *run*. Minimalizaci energie (molekulární statika) provede příkaz *minimize*. Paralelní ohřívání (reklikální výměna) je spouštěna položkou *temper*.

V programu Lammmps můžeme vytvářet modely jednoduchých krystalů (pravidelná struktura). Pokud simulujeme oblast, skládající se z více druhů atomů a mnoha krystalových zrn, je lepší připravit si soubor v jiném, specializovaném programu. Existuje možnost modelovat různé druhy namáhání předepsané tlakem nebo silou, kontakty těles (tření), proud částic, rozpouštění látek a podobně. Dokonce se dají vytvářet jednodušší molekuly s rozpadajícími se vazbami (vlivem namáhání). Bohužel program Lammmps neumožňuje simulovat chemické reakce různých prvků (vznik kovalentních vazeb).

Dále je uveden text, který lze zkopírovat do vstupního souboru (samozřejmě bez přidaných komentářů) a podle něj se provede tah krystalu.

3d simulace tahu kovu

units metal

styl *metal*, nastaví jednotky:

- ✓ [vzdálenost] = Å
- ✓ [čas] = pikosekundy
- ✓ [síla] = elektronvolt/ Å
- ✓ [teplota] = stupně K
- ✓ [tlak] = bary

Výchozí nastavení časového kroku

- ✓ $dt = 0.001$ ps

boundary s s s

- ✓ Určí způsob, jakým se budou chovat hranice simulované oblasti, každé písmeno nastavuje vlastnosti vždy dvou rovnoběžných rovin hranice s normálami směrů x , y a z .
- ✓ s – Je typem neperiodické hranice, která se chová jako smršťovací fólie. Neperiodická v tomto případě znamená, že částice na protilehlých stranách simulované oblasti nemohou přes tuto hranici interagovat, ani ji přecházet. Tento typ hranice neomezuje pohyb atomů; pokud by ji měly při svém pohybu překonat, hranice se posune.

atom_style atomic

- ✓ V simulaci lze použít různé částice, například dipóly. Styl „atomic“ určí, že částice budou atomy. Styl nelze během simulace změnit. Druh částic ovlivňuje množství informací uchovávaných pro každou z nich. Použitý typ částic má vliv především na množství informací přenášovaných při výpočtu sil mezi jednotlivými procesory.
- ✓ Styl „atomic“ každému atomu přiřadí souřadnice, rychlost a identifikační číslo.

lattice fcc 3.52

- ✓ Tímto příkazem definujeme typ mřížky. Mřížka určuje způsob umístění atomů v prostoru prostřednictvím primitivní buňky s bázovými atomy.
- ✓ Styl „fcc“ definuje krychlovou buňku s délkami hran ($a_1 = 1.0$ 0.0 0.0, $a_2 = 0.0$ 1.0 0.0 a $a_3 = 0.0$ 0.0 1.0). Buňka obsahuje 4 atomy (tři uprostřed stěn a jeden ve vrcholu).
- ✓ Velikost uvedená za „fcc“ určuje mřížkovou konstantu. Jelikož elementární buňka „fcc“ je krychle, stačí k jejímu geometrickému popisu jediné číslo. V tomto případě se tedy vytvoří buňka s délkou hrany 3.52 angstromů.

region box block 0 5 0 15 0 5

- ✓ Vytvoří oblast s názvem box, o tvaru kvádra, jehož jedna hrana bude úsek na x ose od bodu nula po pět, druhou osu tvoří úsek y osy od nuly po patnáct, podobně i třetí strana bude úsek osy z od nuly do pěti.

create_box 3 box

- ✓ Vytvoří simulační oblast, která bude obsahovat tři typy atomů, v prostoru vymezeném souřadnicemi definovanými v box

**lattice fcc 3.52 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1 &
 origin 0 0 0**

- ✓ „Lattice“ definuje uzlové body, které budou používat další příkazy.
- ✓ Nakopíruje do simulované oblasti mřížku tak, že x osa oblasti bude splývat se směrem (1 0 0) y osa se směrem (0 1 0) elementární buňky a podobně i pro třetí osu.
- ✓ Parametr „origin“ určí, jak bude primitivní buňka posunuta vzhledem k oblasti, do které se má nakopírovat. V tomto případě dojde k jejímu zarovnání jedním rohem na bod [0 0 0]

create_atoms 1 box

- ✓ Příkaz „create_atoms“ vyplní oblast atomy; číslo jedna značí, že při vyplňování oblasti se použije pouze jeden typ atomů
- ✓ Argument „box“ určí, že oblastí bude celý simulační prostor definovaný příkazem „create_box“.
- ✓ Vyplnění atomy probíhá tak, že hranice, které jsou definované jako symetrické, obsahují pouze jeden atom pro obě strany (ze dvou atomů ležících naproti sobě ve dvou rovnoběžných symetrických rovinách je vytvořen pouze jeden).

pair_style eam

- ✓ Určí typ párového potenciálu.

pair_coeff * * Ni_u3.eam

- ✓ Načte koeficienty pro definovaný párový potenciál (ze souboru).

neighbor 0.3 bin

- ✓ Nastavuje hranice pro oblasti, o kterých si procesory při výpočtech vyměňují informace. Pro nepříliš rozsáhlé problémy není tato hodnota důležitá. U větších může výrazněji ovlivnit dobu výpočtu.

neigh_modify delay 5

- ✓ Určuje největší počet časových kroků, během kterých se musí provést příkaz „neighbor“

region lower block INF INF INF 0.1 INF INF

- ✓ Vytvoří nový blok s názvem „lower“

region upper block INF INF 14.9 INF INF INF

region test block INF INF 0.1 14.9 INF INF

group lower region lower

- ✓ Všem atomům z oblasti „lower“ přiřadí skupinu „lower“

group upper region upper

group boundary union lower upper

- ✓ Všechny atomy, které náleží skupině „lower“ a současně „upper“ jsou zařazeny do skupiny „boundary“.

group mobile subtract all boundary

- ✓ Všechny atomy, které nepatří do skupiny „boundary“, ale náleží skupině „all“, jsou přiřazeny skupině „mobile“.

set group lower type 2

- ✓ Nastaví všem atomům ze skupiny „lower“ typ na hodnotu 2 (vstupem pro příkazy provádějící operace s atomy, jako třeba předepsání rychlosti, výpis informací o atomu do souboru, je typ atomů, ne jméno skupiny)

set group upper type 3

compute new3d mobile temp

- ✓ Definiuje způsob, jakým se bude počítat teplota pro skupinu atomů „mobile“.
- ✓ Výsledná kinetická energie se získá z teploty podle vztahu

$$E_k = \frac{sv}{2Nk_bT},$$

kde s_v jsou stupně volnosti hmotného bodu, N je počet atomů, T teplota v Kelvinech a k_b Boltzmannova konstanta.

compute new2d mobile temp/partial 1 0 1

- ✓ Stejný jako předchozí případ, ale pro kinetickou energii je umožněn pohyb pouze ve dvou stupních volnosti „y_flag“ = 0

compute napeti all stress/atom

- ✓ Spočítá napětí působící na jednotlivé atomy.

výpočet rovnovážné polohy

velocity mobile create 300.0 5812775 temp new3d

- ✓ Skupině atomů přiřadí rychlost odpovídající teplotě 300 stupňů. Rychlosti všech atomů nejsou stejné, číslo 5812775 je vstup, pomocí kterého se nastaví generátor náhodných čísel. Teploty se přiřadí jednotlivým atomům na základě náhodně generovaných čísel.

fix 1 all nve

- ✓ Za klíčovým slovem „fix“ následují příkazy, které se provádí během každého časového kroku.
- ✓ Příkaz „ave“ přepočítává rychlost a polohy atomů.

fix 2 boundary setforce 0.0 0.0 0.0

- ✓ Nastaví všechny síly působící ve skupině atomů „boundary“ na nulu.

fix 3 mobile temp/rescale 10 300.0 300.0 10.0 1.0

- ✓ Každých deset kroků výpočtu se provede kontrola a eventuálně přepočet teploty atomů.
- ✓ Pokud je teplota atomů vyšší nebo nižší než tři sta stupňů o víc než deset, přiřadí novou rychlost tak, aby odpovídala teplotě tři sta stupňů.

fix_modify 3 temp new3d

✓ Upraví způsob výpočtu teploty pro atomy definované v „new3d“
thermo 25

- ✓ Určí, jak často se bude provádět výpis termodynamických veličin (do „log.file“ a na obrazovku)

thermo_modify temp new3d

- ✓ Určí způsob, jakým se budou určovat termodynamické veličiny v příkazech ,které je počítají (teplota, tlak, energie).

timestep 0.0001

- ✓ Určí velikost časového kroku pro výpočet.

run 100

- ✓ Spustí výpočet se sto časovými kroky.

Vytvoření tahu

unfix 3

- ✓ Zastaví výpočty provádění příkazem „fix 3“na současně s tím zruší platnost „fix_modify 3“ přepočítávajícího teploty.

velocity upper set 0 1 0

velocity lower set 0 -1 0

- ✓ Nastaví všem atomům ve skupině „upper“ rychlosti 1 (použije relativní souřadnice pro určení rychlosti, tedy jedna mřížková konstanta za pikosekundu – 1000 cyklů).

fix 3 mobile temp/rescale 10 300 300 10.0 1.0

fix_modify 3 temp new3d

dump 1 all atom 100 dump.tah

- ✓ Příkaz „dump“ provádí výpis informací do souboru.
- ✓ Parametr „all“ vybere všechny atomy v simulaci.
- ✓ Parametr „atom“ určí, že do výstupního souboru se budou zapisovat polohy atomu v relativních souřadnicích (poměry stran simulované oblasti)

- ✓ Poslední parametr příkazu „dump“ je jméno souboru, do kterého se bude zapisovat.

**dump 2 all custom 100 dump.napeti tag c_napeti[1] c_napeti[2]&
c_napeti[3] c_napeti[4] c_napeti[5] c_napeti[6]**

- ✓ Vypíše jednotlivé složky tenzoru napětí pro každý atom do souboru „dump.napeti“

dump 3 test custom 100 dump.polohy tag y

- ✓ Vypíše y souřadnice vybrané skupiny atomů do souboru „dump.polohy“

thermo 100

thermo_modify temp new2d

reset_timestep 0

- ✓ Počítadlu časových kroků nastaví hodnotu na 0

run 60000

- ✓ Provede výpočet s 60000 časovými kroky

Základním výstupem programu je soubor obsahující polohu částic. V první fázi je vhodné zobrazit částice v prostoru, abychom získali lepší představu o jejich poloze (představit si polohu necelých dvou tisíc částic v prostoru na základě znalosti jejich souřadnic není příliš jednoduché). Pro zobrazování atomů a molekul je určen program AtomEye. Ten bohužel nepodporuje vstup ze souborů „dump“, které vytváří LAMMPS. K jejich konverzi je vhodné použít program Pizza (nutné nainstalovat python před spuštěním programu) s hotovými procedurami pro převod. Ve spuštěném programu Pizza stačí zadat

```
d = dump("jmeno_dump_vystupniho_souboru")
d.sort()
c = cfg(d)
c.many("jmeno_souboru_ktere_se_vytvori")
```

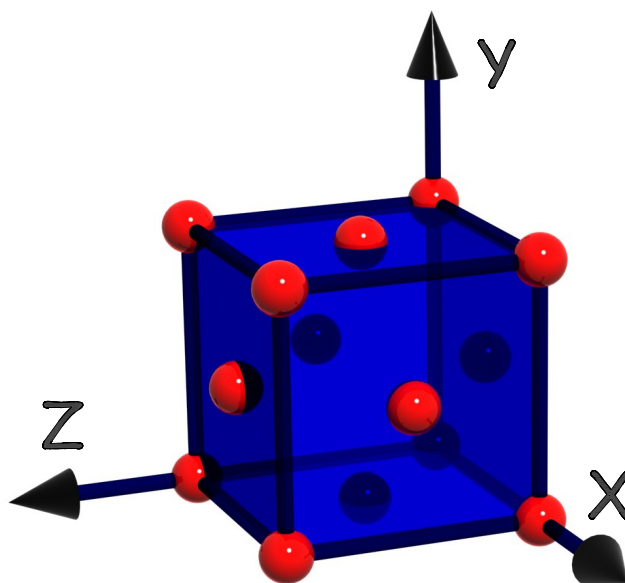
Výše uvedená posloupnost příkazů vytvoří soubory, z nichž každý bude obsahovat informace o polohách atomů v daném časovém okamžiku. Pokud se tak nestane, je potřeba stáhnout poslední verzi programu Pizza. Procedura „cfg()“ byla přidána teprve nedávno. Takto převedené soubory lze otevřít v programu AtomEye. Podle vytvořené animace lze ověřit, zda simulace proběhla správně. Velmi důležité je nezapomenout atomy seřadit podle jejich ID (identifikační číslo). Příkaz „dump“ totiž zapisuje polohy atomů do souboru v pořadí tak, jak přicházejí z jednotlivých procesorů, tedy zcela náhodně. Aby později šlo zjistit, která poloha patří jakému atomu, je každá z nich označena identifikačním číslem. Vizualizační program má trochu jinou logiku vstupních souborů. Místo, aby dohledával atomy podle jejich ID, předpokládá, že každý atom je vždy zapsán ve výstupním souboru ve stejném pořadí. Proto je nutné atomy před vypsáním seřadit procedurou sort().

6 Simulace jednoosého tahu fcc krystalu

6.1 Tah elementární buňky při různých teplotách

Tahová zkouška patří k nejběžnějším experimentálním technikám. Tato kapitola se zabývá teoretickou studií speciálního případu tahového zatížení monokrystalu.

Aby bylo možné porovnávat hodnoty napětí získané ze simulace programem Lammmps s jinými výsledky (získanými pomocí ab-initio výpočtů) byla v modelu vytvořena jediná fcc buňka niklu. Zkratka fcc pochází z anglického Face Centered Cubic (plošně centrovaná kubická). Což je pravidelná krychlová buňka, která má v prostředku každé strany jeden atom (viz obrázek 4) [16]. Atomy jsou zobrazeny červeně, a průniky rovin tvořících povrch jsou zvýrazněny čarami. Její dvě hranice byly simulovány jako periodické (nekonečná deska). Hranici, na které je vytvořen tah, nelze simulovat jako periodickou, protože přitom nejsou přiřazeny stejné vlastnosti atomům ležícím na hranici (vektory jejich rychlostí jsou opačné). Pod a nad taženou buňku bylo přidáno čtyřicet dalších buněk, které simulují „nekonečnou“ délku krystalu ve směru tahu (samozřejmě výsledné napětí se počítá pouze z tažené buňky). Během simulace neprobíhala relaxace vedlejších napětí σ_x a σ_z . Tah je vytvořen předepsáním rychlostí pro atomy na horní a spodní stěně buňky ve směru $[0\ 1\ 0]$ (což je směr osy y), jde tedy o tvrdé zatěžování. Program Lammmps bohužel počítá napětí v jednotkách napětí vynásobené objemem. Proto je ještě potřeba získané hodnoty vydělit objemem atomu.

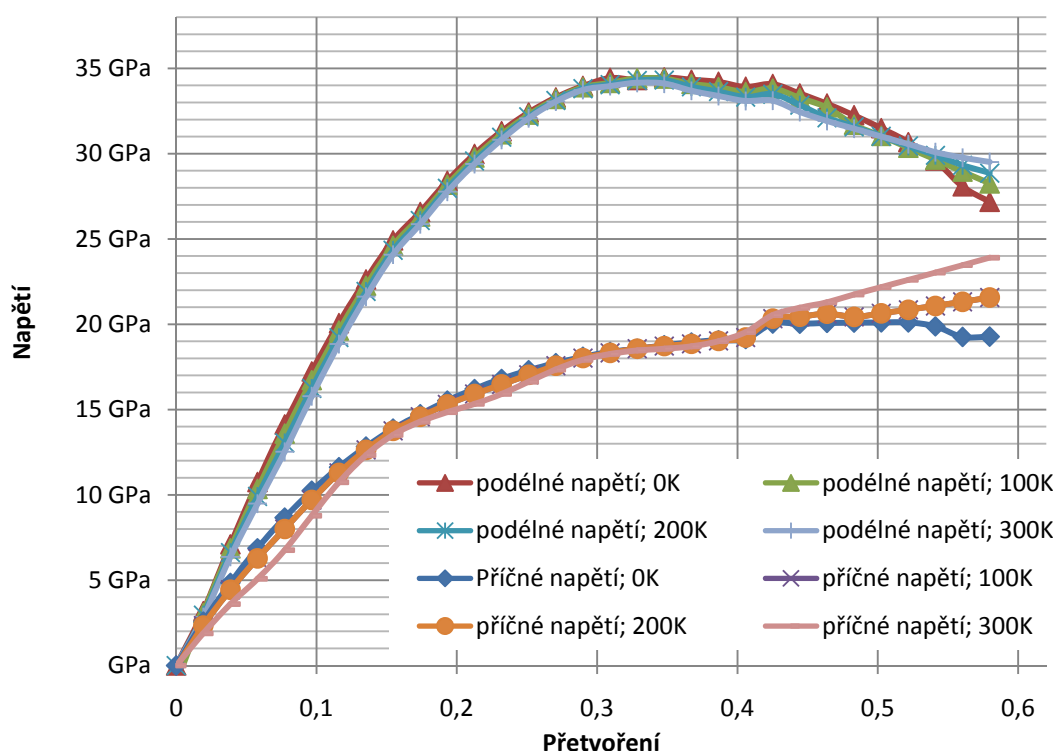


Obrázek 4: Polohy atomů v FCC buňce

Velikost přetvoření se pro tah snadno spočítá podle vztahu

$$\varepsilon = \frac{l - l_0}{l_0},$$

kde l je délka vzorku a l_0 je délka vzorku před deformací. Veličina ε je bezrozměrná.



Obrázek 5: *Tah niklu při různých teplotách*

Napětí ve směru tahu σ_y je v grafu (obrázek 5) označené jako podélné napětí. Pro jednoosý tah platí, že $\sigma_x = \sigma_z$. V grafu jsou tato stejná napětí označena jako příčná.

Z grafu (obrázek 5) je vidět, že velikost maximálního napětí, kterým lze na buňku působit, aniž by se porušila (mez pevnosti), není nikterak významně závislá na teplotě. Tento výsledek na první pohled překvapující je důsledkem toho, že provádíme tah pouze na jediné buňce. Nemohou se tedy projevit různé teplotně aktivované skluzové systémy známé u vzorků konečných rozměrů.

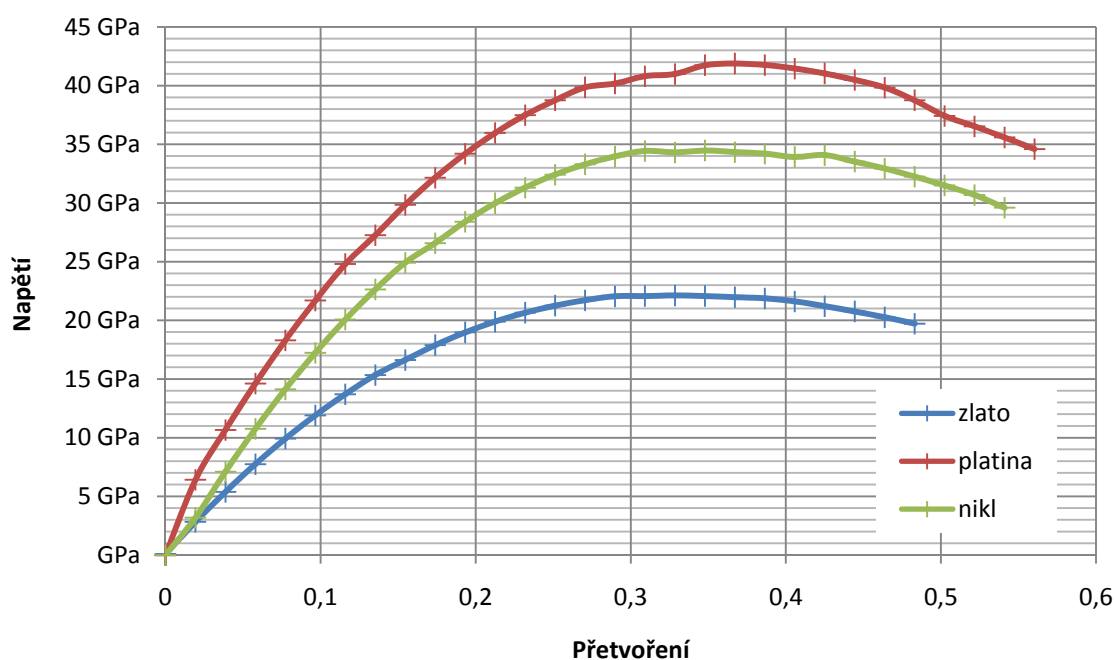
6.2 Tah elementární buňky různých kovů

Taková zkouška se provádí zejména pro určení meze pevnosti materiálu. Jelikož jde o simulaci provedenou při nulové teplotě, nejvyšší hodnota napětí je mezi křehké pevnosti při jednoosé deformaci.

Hodnoty podélných napětí při tahu elementárních buněk jsou vykresleny v grafu na obrázku 6. Tah byl proveden stejným způsobem jako tah buňky niklu v předchozí kapitole. Spočítané hodnoty maximálních napětí podélných a jim odpovídající příčná napětí a přetvoření jsou shrnuty v tabulce 1.

Prvek	podélné napětí (σ_y)	příčné napětí ($\sigma_{x,z}$)	přetvoření (ε_y)
Nikl	34,5 GPa	18,8 GPa	0,34
platina	41,9 GPa	40,8 GPa	0,36
zlato	22,1 GPa	21,7 GPa	0,30

Tabulka 1: Hodnoty při maximálním tahovém napětí σ_y



Obrázek 6: Graf průběhů napětí σ_y v závislosti na přetvoření

Než je možné výsledek označit za správný, je nutné jej nějakým způsobem ověřit, nejlépe experimentem. Jelikož nemáme k dispozici zařízení, na kterém by šlo takové

měření provést, porovnáme námi vypočítané hodnoty alespoň s hodnotami získanými jinou výpočetní metodou a s výsledky, ke kterým dospěli jiní autoři.

Tyto hodnoty byly porovnány s hodnotami uvedenými v [17]. V této práci byla počítána různá napětí pro jednoosou zátěž (tedy při relaxaci příčných napětí), užitím metody vnořeného atomu. Pro největší hodnoty napětí ve směru [1 0 0] (ekvivalentní našemu [0 1 0]) je ve zmiňované literatuře je uvedená velikost maximálního napětí pro nikl 39GPa, při přetvoření 0,327 a pro zlato pro stejný směr zatěžování je maximální napětí 22,5GPa a přetvoření 0,188.

V porovnání s hodnotami uvedenými v práci [17] se napětí vypočítané programem Lammmps u maximálního napětí zlata liší zhruba o 2% a niklu o 12%. Je velmi zvláštní, že pro jeden prvek je shoda výsledků relativně dobrá a pro jiný naopak velice špatná. Na rozdíl od výpočtů v [17], v našich výpočtech nebyla provedena Poissonova kontrakce (její neprovedení zapříčiní vznik příčných napětí). Vzniklá příčná napětí ovlivní velikost maximálního tahového napětí u některých prvků více a u jiných méně. Přetvoření pro nikl vyšlo o 6,3% větší a pro zlato větší dokonce o 64%.

Získané hodnoty byly dále porovnány s hodnotami ab-initio výpočtů, přičemž hodnota maximálního napětí pro nerelaxovaný tah byla pro nikl spočítána metodou ab-initio na 34,6 GPa při přetvoření 0,32. Což je velmi dobrá shoda s údaji získanými ze simulace molekulární dynamiky. Maximální napětí spočítané pomocí MD je o 0,4% menší a přetvoření o 6,3% větší, než hodnoty spočítané metodou ab-initio.

Bohužel nemáme k dispozici hodnoty maximálních napětí při nerelaxovaném tahu zjištěné přímo z ab-initio výpočtů. Jak je uvedeno v práci [18], existuje pro některé kovy lineární závislost mezi maximálním napětím a superponovaným příčným napětím. Pokud tedy známe maximální napětí relaxovaného tahu a příčná napětí nerelaxovaného, můžeme snadno podle tabelovaných konstant určit zbývající podélné napětí nerelaxovaného tahu, podle vztahu

$$\sigma_{\max} = \sigma_{\max,0} + k_{\max} \sigma_{př},$$

kde $\sigma_{\max,0}$ je maximální napětí jednoosé napjatosti v tahu, k_{\max} je konstanta úměrnosti pro daný prvek a $\sigma_{př}$ je příčné napětí při jednoosé deformaci. Konstanty k_{\max} jsou převzaté z [18]. Dále byly z grafu uvedeného v [18] určeny velikosti přetvoření při maximálním napětí.

✓ pro zlato:

$$\sigma_{y\max} = 18,6 + 0,163 \cdot 21,735 = 22,1 \text{ GPa}$$

$$\varepsilon_{y\max} \doteq 0,24$$

Napětí je o 0,3% větší, než námi určená hodnota pomocí MD

Velikost přetvoření vyšla o 20% menší, než hodnota spočítaná pomocí MD

✓ pro platinu:

$$\sigma_{y\max} = 34,1 + 0,152 \cdot 40,83 = 40,306 \text{ GPa}$$

$$\varepsilon_{y\max} \doteq 0,23 \text{ (hodnota určená extrapolací z grafu)}$$

Hodnota napětí je o 3,8% menší, než námi spočítaná hodnota pomocí MD

Velikost přetvoření vyšlo o 36% menší, než přetvoření spočítané metodou MD

Spočítaná napětí se ve všech případech docela dobře shodují s maximálními napětími určenými jinými autory, bohužel při výpočtu přetvoření už tak dobrá shoda není. Jedním z možných důvodů tak velkého rozdílu u výsledků přetvoření je nepříliš jednoznačný extrém napětí. Křivka popisující průběh napětí v závislosti na přetvoření je v blízkosti svého maxima velmi plochá (má malou derivaci).

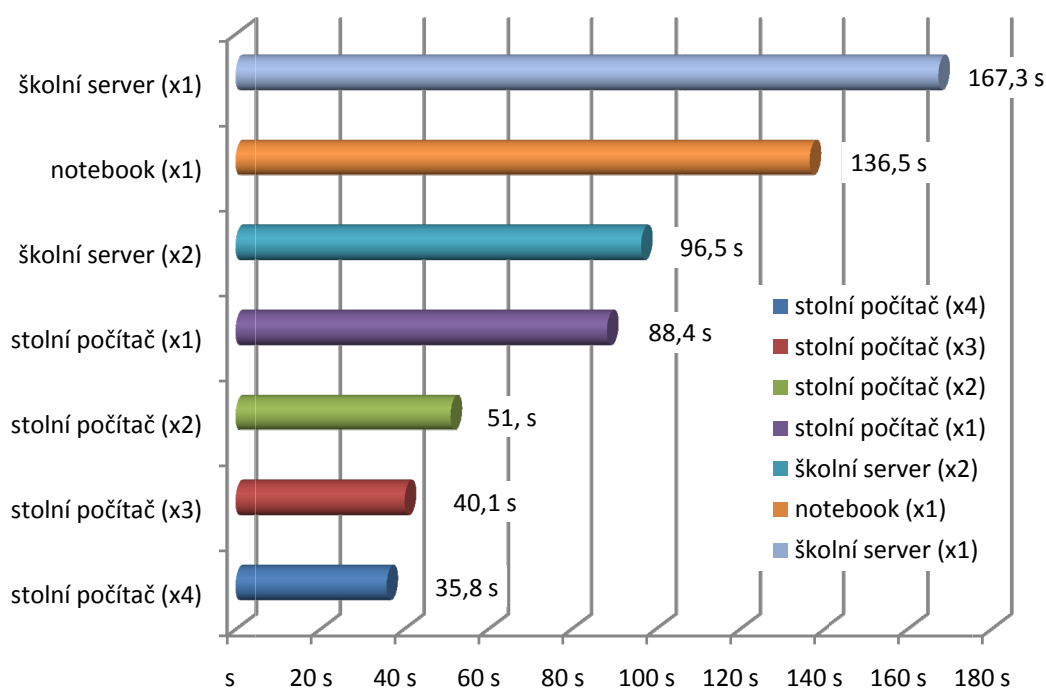
6.3 Tah vzorku konečných rozměrů

Pro další výpočty byla připravena poněkud složitější simulace. Byl vytvořen vzorek tvaru kvádrů $15 \times 5 \times 5$ elementárních FCC buněk, jehož stěny tvoří hranice krystalu. Simuluje se tedy o tah „nanodrátka“ (s čtvercovým průřezem) o rozměrech $1,76\text{nm} \times 1,76\text{nm} \times 5,28\text{nm}$. Tah je aplikovaný na nejmenší stěnu kvádrů. Tato simulace byla pro srovnání provedena na několika typech počítačů. Jelikož jde o systém s malým počtem atomů, byl spuštěn vždy jen na jednom procesoru. Použití více procesorů by bylo neefektivní, vzhledem k náročnosti komunikace mezi jednotlivými procesy (jádra jednoho procesoru jsou na sběrnici, jejíž propustnost je mnohonásobně vyšší než propustnost sběrnice mezi procesory nebo propustnost sítě, která spojuje jednotlivé počítače). Paměť potřebná pro provedení této simulace je zhruba 3 MB. Do výpočtu vstoupilo 1936 atomů, pro které bylo spočítáno celkem 16 000 iterací.

Vždy stejnou úlohu řešily postupně tři počítače. Byly vybrány tak, aby reprezentovaly různé typy počítačů. Stolní počítač byl vybaven nejběžnějším typem procesoru, tedy Intel Core 2 Quad Q6600 2,40GHz 8MB, takt paměti v tomto počítači je 800 MHz. Notebook byl osazen procesorem Intel Core 2 Duo T2370 2,00GHz 2MB, s pamětí pracující s taktem 600 MHz. A jako poslední je vzdálený server, což byl školní počítač s procesorem Pentium D. Pro rychlost výpočtu simulace námi provedeného rozsahu se zdá být velmi důležitá velikost L2 paměti integrované na procesoru. Pokud je

velikost této paměti srovnatelná s pamětí, která je pro výpočet potřeba, doba pro výpočet podstatně klesá. Další velmi významný faktor je počet jader, která daný procesor obsahuje. Na procesoru se dá spustit maximálně takový počet procesů, jako je počet jader (respektive i vyšší, ale způsobí to snížení výkonu namísto zvýšení). Naměřené výsledky jsou shrnuté v grafu na obrázku 10. Údaj v závorce (xN) označuje počet spuštěných procesů na daném procesoru.

Jak už bylo zmíněno výše, vypočítat napětí působící na atom z výstupu programu Lammmps není příliš jednoduché. Proto jsem si vytvořil program, který sečte složky tenzorů napětí rozpočítané programem Lammmps na jednotlivé atomy a ten pak dělí jejich objemem. Objem atomů získá tak, že bere část celkového objemu simulované soustavy v poměru počet atomů, ze kterých určujeme napětí, k celkovému počtu atomů. Samozřejmě tato metoda není úplně přesná, vzhledem k tomu, že při deformaci nebudou mít všechny atomy stejný objem, ale pokud je poměr atomů (počítané/celkem v simulaci) blízký jedné, je takto vzniklá chyba zanedbatelná.

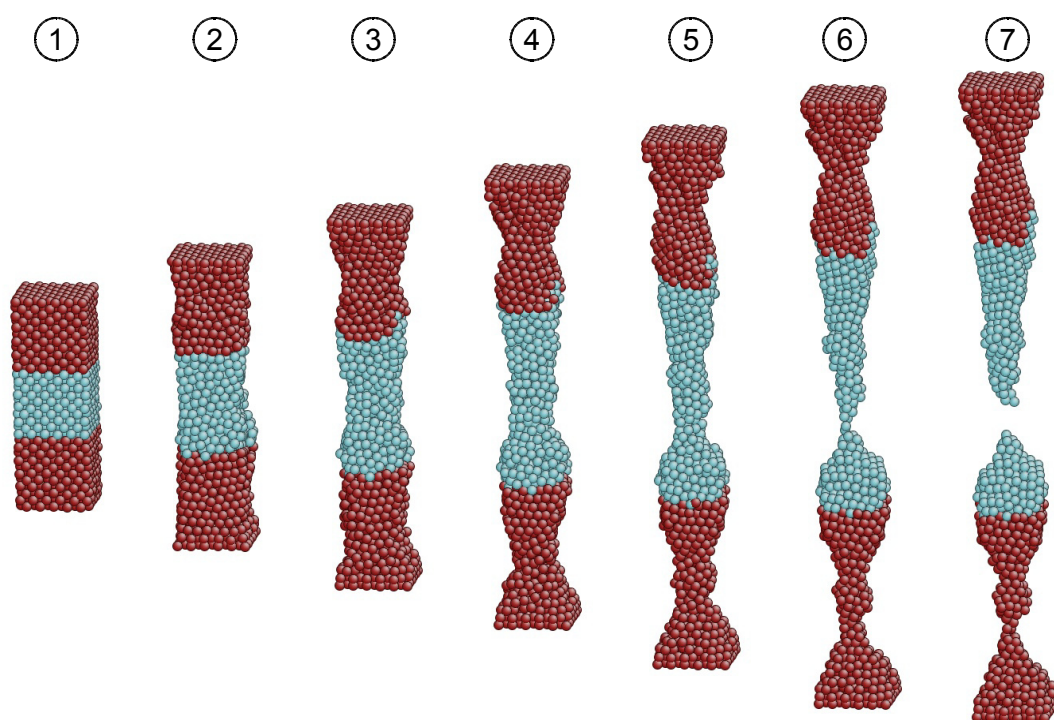


Obrázek 10: Doba výpočtu v závislosti na použitém počítači a počtu spuštěných procesů.

Z dat naměřených pro různý počet jader je patrné, že rozdělení úlohy mezi více než dva procesory již není příliš výhodné (malý nárůst výkonu). Samozřejmě tento závěr

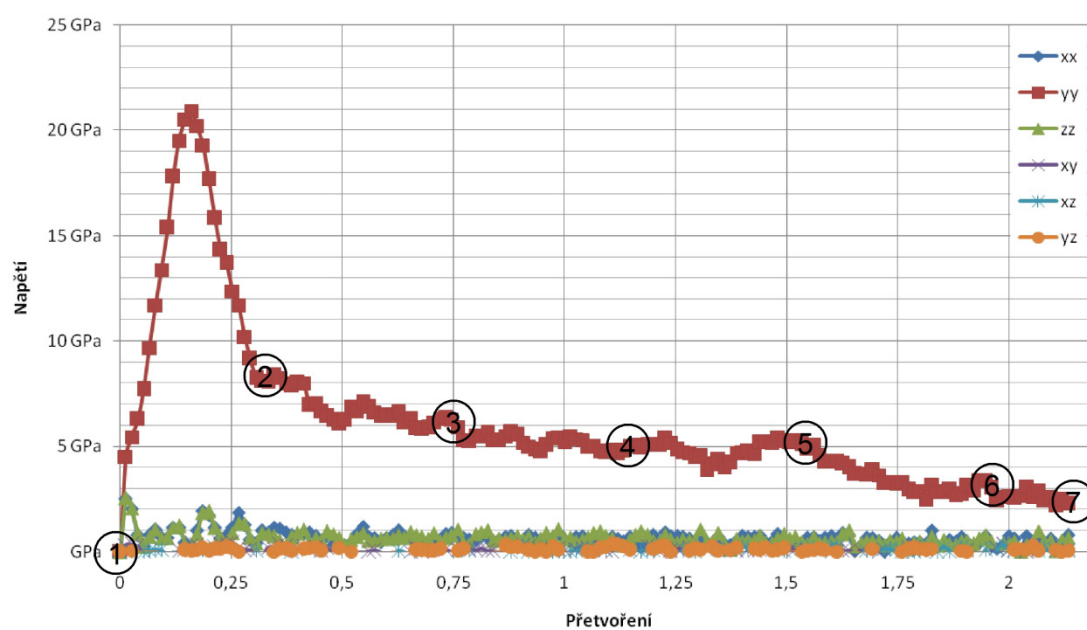
platí pouze pro danou úlohu, pokud by se jednalo o rozsáhlejší problém – stovky tisíc atomů - zaznamenal by se nepochybně výrazný nárůst výkonu při každém zvýšení počtu procesů použitých pro výpočet.

Program AtomEye nabízí možnost, jak sledovat pohyby atomů, v průběhu deformace. Z takto vytvořeného videa bylo vybráno několik obrázků zachycujících deformaci v taženém vzorku (obrázek 11). Číslo u vzorku slouží k informaci o přetvoření, kterou lze odečíst z grafu na obrázku 12. Na tomto grafu jsou vyznačeny průběhy složek tenzoru napětí v závislosti na přetvoření.



Obrázek 11: *průběh přetváření vzorku z Ni při tažení.*

Mohlo by se zdát zvláštní, že po roztržení vzorku neklesne napětí ihned na nulu (obrázek 12). To je důsledkem vnitřních pnutí, která se uvolňují velmi pomalu. Dále je patrné, že maximální napětí v taženém vzorku (mez pevnosti) je daleko nižší, než mez pevnosti nekonečně velkého krystalu.



Obrázek 12: průběh napětí při přetváření vzorku z niklu tažením.

6 Závěr

Podařilo se přeložit a zprovoznit program Lammmps v linuxovém prostředí, pomocí něhož byla později provedena simulace tahu vzorku z niklu. K výpočtu výsledného napětí ve vzorku jsem vytvořil jednoduchý program v pascalu, který používá jako vstup pro výpočet napětí soubory obsahující informace o tenzoru napětí přepočítané pro jednotlivé atomy a termodynamický soubor, ze kterého je načítán objem simulované oblasti.

Byla provedena simulace jednoosé deformace (bez Poissonovy kontrakce) tří vybraných fcc kovů při nulové teplotě. Na základě spočítaných hodnot byla určena mez křehké pevnosti nekonečného krystalu pro zlato 22,1 GPa, pro nikl 34,5 GPa a pro platinu 41,9 GPa. V případě niklu byla tato simulace provedena i při vyšších teplotách. Z výpočtů vyplynulo, že nízké teploty (do 300 K) téměř neovlivní mez pevnosti nekonečného krystalu. Nakonec byla provedena simulace tahové zkoušky vzorku konečných rozměrů z niklu, při pokojové teplotě. Maximální napětí ve zmiňovaném vzorku bylo 21 GPa, tedy 1,6 krát menší než v nekonečně velkém krystalu ze stejného materiálu.

7 Použitá literatura

- [1] Alder B., Wainwright T., Studies in Molecular Dynamics. I. General Method, Phys, 459 s., 1959,
- [2] Pokluda J., Kroupa F., Obdržálek L., Mechanické vlastnosti a struktura pevných látek: kovy – keramika – plasty, 1. Vyd. Brno, PC-DIR spol. s.r.o., 386 s., ISBN 80-214-0575-9
- [3] Daw S., Baskes I., Phys. Rev. B 29, 1984
- [4] Broughton J., Bristowe P., Newsam J., Materials Theory and Modelling, MRS Proceedings 291, 37 s., 1993,
- [5] Mendeleev M., Srolovitz D., Physical Review B, APS, 2002
- [6] Wallenius J., Abrikosov I., Chakarova R., Journal of Nuclear Materials, Vol. 329-333, pp. 1175-1179, 2004
- [7] Rottach R., Curro G., Budzien J., Grest S., Svaneborg C., Everaers R., Molecular dynamics simulations of polymer networks undergoing sequential cross-linking and scission reactions, 40 s., 2007
- [8] Linux Magazine, <http://www.linux-mag.com/>, 2008-04
- [9] Theoretical and Computational Biophysics Group, <http://www.ks.uiuc.edu/Research/vmd/>, 2008-04
- [10] Nvidia, http://www.nvidia.com/object/IO_43499.html, 2008-04
- [11] Wikipedie: Otevřená encyklopedie, <http://cs.wikipedia.org/wiki/>, 2008-04
- [12] Nvidia: Cuda Zone, http://www.nvidia.com/object/cuda_home.html, 2008-04
- [13] Lexico Publishing Group, <http://dictionary.reference.com/> 2008-03
- [14] Teufel S., Linux a KDE: Podrobný průvodce, Grada, 260 s. ISBN 80-7169-944-6.
- [15] Lammmps, <http://lammmps.sandia.gov/doc/> 2008-05
- [16] Stránský L., Krystalická stavba kovů, Doplnující podklady pro cvičení
- [17] Milstein F. Chantasiriwan S., Theoretical study of the response of 12 cubic metals to uniaxial loading, Phys. Rev. B 58, pp. 6006–6018, 1998
- [18] Černý M., Pokluda J., Influence of superimposed biaxial stress on the tensile strength of perfect crystals from first principles, Phys. Rev. B 76, pp. 024115-24119, 2007
- [19] Hinchliffe A., Molecular Modelling for Beginners, WILEY, ISBN 0-470-84310-1
- [20] Halliday D., Resnick R., Walker J., Fyzika, VUTIUM, ISBN 80-214-1868-0

8 Použité symboly a zkratky

X - označuje skalární veličinu, nebo velikost vektorové veličiny

\vec{X} - označuje vektorovou veličinu

F - síla

U - potenciální energie

\vec{E} - Intenzita elektrického pole

ε_0 - relativní permeabilita

$$- \varepsilon_0 = 8.854 \cdot 10^{-12} \cdot C^2 \cdot N^{-1} \cdot m^{-2}$$

Q, q - Velikost náboje

ε - přetvoření

σ - napětí

\vec{r} - Polohový vektor

π - Ludolfovo číslo ($\doteq 3.14$)

\AA - jednotka používaná zejména v krystalografii $1 \cdot 10^{-10} m$

MD - molekulární dynamika

EAM - metoda vnořeného atomu (embedded atom method)

L-J - Lenardův Jonesův potenciál (druh párového potenciálu)

FCC - typ krystalové buňky, plošně středěná kubická